

## CLAIMS

What is claimed is:

1. A method for managing state data of a service in a service-oriented architecture by establishing a gateway for service-oriented state comprising:

configuring an extensible, pluggable interface to support for extensible processor interfaces; data query support on service state data, automated notification capability on service state to a client; and automated data transform on service state data to a client format; defining an interface framework for interaction between a service and said gateway;

establishing an extensible meta-data definition comprising an extensible set of service state data attributes including state data qualifiers, constraints, and access mechanisms; and

utilizing one or more pluggable processors configured to utilize said extensible meta-data definition for interfaces and decision making based on said meta-data.

2. The method of Claim 1 wherein said establishing meta-data definition for state data attributes including service state data qualifiers includes:

enabling expression of state change notification qualifiers, said qualifiers including notification of a change in state results;

enabling definition and expression of security requirements on a service state discovery and notification;

enabling expression of transaction requirements, said transaction requirements including enable/disable transaction; and

providing a framework to link to a transaction coordinator and define transaction scope said transaction scope including a service developer controlled transaction, a service controlled transaction, and a gateway managed transaction on state data.

3. The method of Claim 1 wherein said establishing meta-data definition for state data attributes including service state data constraint comprises;

- enabling expression of constraints on mutability;
- enabling expression of constraints on a validity of said service state data through life time constraints, said life time constraints including at least one of, time to live (TTL), time to activate, available from, available until, good from, good until and immutable until;
- enabling cardinality on said service state data; and
- enabling and specification of a relationship among said service state data, said specification supporting complex queries and state management.

4. The method of Claim 1 wherein said establishing meta-data definition for state data attributes including service state data access mechanisms comprising:

- enabling a flexible callback mechanism on said service state data and expression thereof through meta-data;
- establishing a data push mechanism for service state data update with expression of said constraints through meta-data;
- establishing other extensible data access mechanisms on said service state data, said other extensible data access mechanisms including direct access to said service state data in a database; direct access to said state data through SNMP, CIM, Web services; and
- establishing an extensible custom template mechanism for service state data access based on service requirements; said service state data access based on service requirements including a service template for custom scripts such as SQL.

5. The method of Claim 1 wherein said extensible data query support on service state data includes;

enabling a service developer to define a query type based on a state data schema definition;

enabling a service user to send a state data query and query type to said service and transmitting said state data query and query type to a service state query processor; wherein said service state query processor evaluates said query and informs said gateway with state data information to facilitate processing said query and said gateway retrieves said state data information using meta-data information of said service state data;

wherein said state data information is converted to a canonical data format to facilitate comprehension by said service query processor;

wherein said query processor conducts said query on said service state data; and

wherein query results are sent back to said client in a format as requested using a transformation processor.

6. The method of Claim 1 wherein said automated notification capability on service state to a client includes:

enabling said gateway to define a pluggable framework for a notification processor based on notification semantics, said notification semantics including selected query criteria established by said client;

enabling said gateway to send notification to a client based on a state change, wherein said state change is a result of at least one of a state data push model, a state data pull model, and another state change indication; and

said notification processor is configured to utilize a transformation processor to facilitate providing state data to a client in a selected format.

7. The method of Claim 1 wherein said automated data transform on service state data to a client format comprises:

defining a transformation processor configured to transform results of a query or notification on state data based on a client requirement;

enabling a client to specify a format for results, said gateway formatting said results using an appropriate transformation processor; and

wherein a query processor and a notification processor are configured to employ said transformation processor.

8. The method of Claim 1 wherein said extensible meta-data definition for interfaces facilitates providing said client with meta-data and semantic information comprising:

obtaining a current set of state meta-data on said service;

revising meta-data about said service state data;

establishing consistent meta-modeling techniques; and

providing meta-model versioning capabilities and version compatibility to support any meta-data.

9. The method of Claim 1 further including providing a framework for meta-data, and a meta-model repository to support caching capability on said service state data execution.

10. The method of Claim 9 wherein said framework includes at least one of an in-memory XML DOM representation for XQuery and a Java object caching mechanism.

11. The method of Claim 1 further including defining a meta-data language, said language comprising an annotation to existing service state data schema and a reference to said service state data and its schema.

12. The method of Claim 1 further including defining a flexible meta-model definition for meta-data.

13. The method of Claim 1 wherein said flexible meta-model definition supports different versions of meta-data and enables consistency across meta-data modeling.

14. The method of Claim 1 further including supporting a meta-data schema processor with extensibility to support new meta-data extensions including a semantic information model on said service state data.

15. The method of Claim 1 further including providing persistence storage for said service state data and its meta-data, said storage facilitating recoverability and scalability on said service state data and associated meta-data.

16. The method of Claim 1 wherein said interface framework includes at least one of an interface for managing meta-data associated with said service state data, an interface to facilitate access to gateway functionalities including query execution, semantics and to access a gateway information model, and a plug-in architecture to manage meta-data, query, notification and transform processors.

17. A system for managing state data of a service in a service-oriented architecture by establishing a gateway for service-oriented state comprising:

a means for configuring an extensible, pluggable interface to support for extensible processor interfaces; data query support on service state data, automated notification capability on service state to a client; and automated data transform on service state data to a client format; defining an interface framework for interaction between a service and said gateway;

a means for establishing an extensible meta-data definition comprising an extensible set of service state data attributes including state data qualifiers, constraints, and access mechanisms; and

a means for utilizing one or more pluggable processors configured to utilize said extensible meta-data definition for interfaces and decision making based on said meta-data.

18. A storage medium encoded with a machine-readable computer program code, said code including instructions for causing a computer to implement a method for managing state data of a service in a service-oriented architecture by establishing a gateway for service-oriented state, the method comprising:

configuring an extensible, pluggable interface to support for extensible processor interfaces; data query support on service state data, automated notification capability on service state to a client; and automated data transform on service state data to a client format; defining an interface framework for interaction between a service and said gateway;

establishing an extensible meta-data definition comprising an extensible set of service state data attributes including state data qualifiers, constraints, and access mechanisms; and

utilizing one or more pluggable processors configured to utilize said extensible meta-data definition for interfaces and decision making based on said meta-data.